D4

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 694 846 A1

(12) DEMANDE DE BREVET EUROPEEN

(43) Date de publication:
31.01.1996 Bulletin 1996/05

(51) Int Cl.6: G06F 12/14, G06F 7/00, H01L 23/535

(21) Numéro de dépôt: 95401789.3

(22) Date de dépôt: 27.07.1995

(84) Etats contractants désignés:
DE FR GB IT

(30) Priorité: 29.07.1994 FR 9409485

(71) Demandeur:
SGS-THOMSON MICROELECTRONICS S.A.
F-94250 Gentilly (FR)

(72) Inventeurs:
- Wuidart, Sylvie, Cabinet Ballot-Schmit
  F-94230 Cachan (FR)
- Sourgen, Laurent, Cabinet Ballot-Schmit
  F-94230 Cachan (FR)

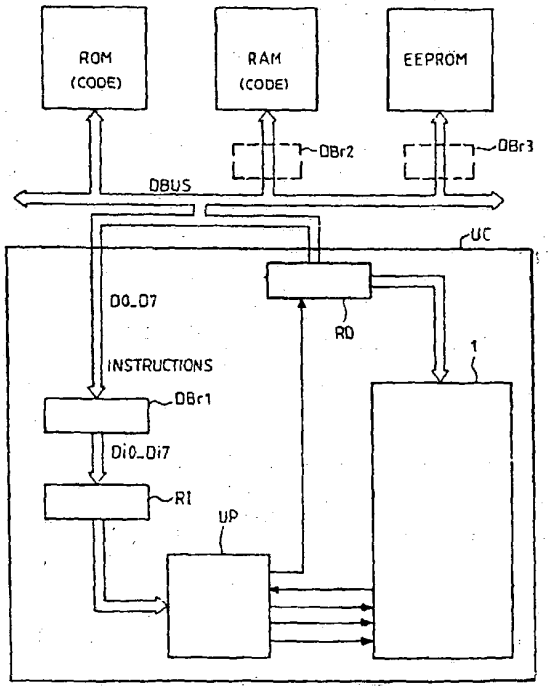
(74) Mandataire:
Schmit, Christian Norbert Marie et al
F-94230 Cachan (FR)

(54) Procédé de brouillage numérique et application à un circuit programmable

(57) L'invention concerne un procédé de brouillage numérique par permutation des bits de données dans un circuit programmable comprenant une unité de commande et au moins un bus de données pour faire transiter des données entre l'unité de commande et différentes mémoires. Les données transitent soit en clair, soit brouillées sur ledit bus de données selon qu'elles sont des instructions ou non, et les données mémorisées dans certaines de ces différentes mémoires sont brouillées.

L'invention concerne aussi un procédé de réalisation du circuit de permutation.

FIG_2



EP 0 694 846 A1

Description

L'invention concerne un procédé de brouillage numérique et son application à un circuit programmable, essentiellement pour protéger de l'inspection le contenu d'un programme exécutable. L'invention s'applique notamment aux circuits sécurisés de gestion de transactions financières.

Les données d'instructions d'un programme contenu dans une mémoire programme sont habituellement protégées par mélange des adresses physiques en mémoire de ces données d'instructions. Ainsi, ces données ne sont plus rangées dans la mémoire selon leur adresse logique, mais éparpillées. En mode opérationnel du circuit programmable, un circuit de décodage spécifique permet de retrouver l'adresse physique réelle en mémoire d'une donnée d'instructions d'après son adresse logique.

Ce procédé permet une protection contre l'inspection visuelle de la mémoire programme. Cependant, les données d'instructions transitent en clair sur le bus de données, pour être exécutées par le microprocesseur du circuit. Or, il est techniquement possible de placer un espion sur le bus de données, pour lire les données qui transitent. Le programme peut alors être reconstitué. En pratique la protection apportée est donc limitée.

En outre, la complexité du circuit de décodage d'adresse permettant de retrouver les adresses physiques réelles conduit à utiliser les mêmes clefs de codage et partant, le même circuit de décodage pour une famille de produits donnée, afin de réduire les coûts de fabrication.

Or, si un fraudeur arrive à reconstituer le programme pour un des produits, il obtient une correspondance entre les adresses physiques et les adresses logiques, ce qui lui permettra de retrouver les clefs et l'algorithme utilisés pour mélanger les adresses. Il aura ainsi la solution pour toute la famille.

Un objet de l'invention est d'améliorer la protection des données dans un circuit programmable en utilisant un procédé de brouillage par permutation de bits de données.

Un autre objet de l'invention est un procédé de protection de ces données facile à mettre en oeuvre, permettant une protection individualisée pour chaque produit.

On sait qu'un programme exécutable contient des données dont certaines constituent des instructions de ce programme.

On prévoit selon l'invention que ou bien ces données d'instructions ou bien les autres données sont brouillées dans la mémoire programme. C'est à dire que la permutation des bits de l'un de ces groupes de données est effectuée à la génération du code exécutable. De cette manière, on a en mémoire programme des données brouillées parmi d'autres données qui ne le sont pas.

On rappelle que selon l'acception courante, on entend par mémoire programme, l'ensemble des adresses physiques auxquelles sont mémorisées des données du code exécutable. Ces adresses peuvent être dans des circuits mémoire différents, par exemple en mémoires ROM et RAM.

Telle que caractérisée, l'invention concerne donc un procédé de brouillage numérique de données.

Selon l'invention, le brouillage est réalisé par un générateur de code exécutable qui effectue une discrimination des données d'instructions parmi toutes les données d'un programme, pour appliquer une première permutation de bits ou bien à ces données d'instructions ou bien aux autres données.

On obtient un code avec des données brouillées mélangées avec des données non brouillées. De cette manière, on améliore considérablement la protection du programme.

Le code exécutable brouillé obtenu est chargé en mémoire programme dans un circuit programmable comprenant une unité de commande.

De préférence, ce sont les données d'instructions qui sont brouillées. Ceci permet avantageusement que les données d'instructions ne soient débrouillées qu'en entrée instruction de l'unité de commande, de manière à transiter brouillées sur le bus de données.

Si ce sont les autres données qui ont été brouillées, elles ne sont avantageusement débrouillées qu'en entrée de données de l'unité de commande. Ou bien une permutation inverse est appliquée en sortie de la mémoire programme en sorte que les données d'instructions transitent brouillées sur le bus de données parmi les autres données en clair. Ces données d'instructions seront alors débrouillées en entrée d'instruction de l'unité de commande. On revient au cas précédent.

Dans un perfectionnement, on peut prévoir d'appliquer une deuxième permutation de bits à toutes les données contenues dans une mémoire réinscriptible de la mémoire programme ou non. En appliquant les données de cette mémoire réinscriptible, à un circuit de permutation placé entre ladite mémoire et le bus de données, on effectue un brouillage des données à mémoriser et un débrouillage des données à transmettre selon cette autre permutation.

De cette manière on obtient des données en mémoire programme brouillées selon la première permutation, voire de la deuxième, mélangées avec d'autres données mémorisées en clair ou brouillées selon la deuxième permutation, tandis que sur le bus transitent des données brouillées selon la première permutation parmi d'autres données en clair. Il devient très difficile pour un fraudeur de s'y retrouver.

L'invention concerne aussi un circuit programmable comprenant au moins un circuit de permutation des bits de

données, pour des données de n bits.

Selon l'invention, le circuit de permutation comprend n étages de permutation à n entrées et une sortie, chaque étage comprenant n cellules de permutation.

Avantageusement, une cellule de permutation est réalisée selon la même technologie que les cellules de la mémoire programme.

D'autres caractéristiques et avantages de l'invention sont présentés dans la description suivante, faite à titre indicatif et non limitatif, et en référence aux dessins annexés dans lesquels :

- la figure 1 représente le principe du brouillage selon l'invention,
- la figure 2 est un schéma-bloc d'un circuit programmable protégé selon l'invention,
- la figure 3 est un schéma de principe d'un circuit de permutation selon l'invention,
- la figure 4 est un dessin topologique de deux étages d'un circuit de permutation selon l'invention et
- la figure 5 est une vue en coupe AA du dessin topologique de la figure 4.

Selon l'invention, un générateur de code exécutable discrimine les données d'instruction des autres données d'un programme, pour appliquer un brouillage par permutation des bits ou bien à ces données d'instructions, ou bien aux autres données. C'est ce qui est représenté sur la figure 1. On obtient un code exécutable qui contient des données brouillées parmi d'autres qui ne le sont pas.

Ce code exécutable est mémorisé en mémoire programme d'un circuit programmable : programmation par masque en mémoire morte et/ou téléchargement en mémoire réinscriptible (RAM, EEPROM ou autres).

Un circuit programmable selon l'invention est représenté en figure 2. Il comprend une unité de commande UC qui est un organe du type microprocesseur, avec une entrée d'instructions sur un registre d'instruction RI et une entrée de données sur un registre de données et qui comprend une unité programmée UP. Cette unité programmée reçoit des données d'instructions à exécuter du registre d'instruction RI et contrôle le registre de données RD et par exemple des circuits de calcul arithmétique et logique, ou des compteurs, regroupés dans un circuit de calcul référencé 1 sur la figure. Dans l'exemple représenté, le circuit de calcul 1 reçoit des données du registre de données RD sur lesquelles il effectue des calculs selon les instructions de l'unité programmée.

Les différentes données sont transférées vers les registres de données ou d'instructions par un bus de données DBUS connecté à différents circuits mémoire.

Dans l'exemple représenté à la figure 2, le circuit programmable comprend une mémoire morte ROM et des mémoires réinscriptibles, dans l'exemple une mémoire vive RAM et une mémoire programmable et effaçable électriquement EEPROM.

Dans cet exemple, on prévoit que du code exécutable est mémorisé en mémoires ROM et RAM.

Selon l'invention, on prévoit que les données de la mémoire programme transitent soit en clair, soit brouillées sur le bus de données selon qu'elles sont des données d'instructions ou non.

De préférence, on choisi que ce sont les données d'instructions qui transitent brouillées. Et un circuit de permutation DBr1 est placé en entrée d'instructions de l'unité de commande, pour débrouiller les données instructions contenues dans la mémoire programme et transférées par le bus de données DBUS. Les données d'instructions transitent donc brouillées sur le bus de données.

Si ce sont les autres données qui ont été brouillées par le générateur de code, une permutation inverse est appliquée en sortie de mémoire programme, en sorte que les données d'instructions sont brouillées sur le bus. Ou bien les données brouillées transitent brouillées sur le bus et un circuit de permutation est prévu en entrée de données de l'unité de commande.

Le procédé selon l'invention permet donc de nombreuses combinaisons.

Dans l'exemple représenté à la figure 2, qui s'adresse plus particulièrement au cas où les données d'instructions transitent brouillées sur le bus parmi les autres données en clair, on a prévu deux autres circuits de permutation DBr2 et DBr3. Le premier est placé entre la mémoire RAM et le bus de données, et le second est placé entre la mémoire EEPROM et le bus de données. De cette manière, les données mémorisées dans ces deux mémoires réinscriptibles sont brouillées respectivement selon une seconde et troisième permutation, mais transitent débrouillées sur le bus de données.

Plus précisément, dans l'exemple représenté, les données d'instructions mémorisées en mémoire RAM ont subi deux permutations : la première qui leur est spécifique, réalisée à la génération du code et la deuxième appliquée en entrée de la mémoire. Et la deuxième permutation est débrouillée avant la transmission des données sur le bus. La deuxième permutation est de préférence différente de la première appliquée aux données d'instructions.

On pourrait aussi avoir une autre mémoire sans circuit de permutation associé.

Ainsi, le procédé selon l'invention permet d'appliquer un brouillage différent aux données d'instructions de la mémoire programme que celui appliqué aux autres données: formes mémorisées, formes transmises différentes, et aussi d'appliquer un brouillage différent à chacune des mémoires réinscriptibles: permutation ou non, sélection de la permutation des bits différente d'une mémoire réinscriptible à l'autre.

Selon l'invention, on accroit ainsi les difficultés d'espionnage des données et de compréhension du dispositif de protection.

La structure retenue dépendra principalement de l'application visée. Enfin, on verra que l'on peut facilement utiliser des permutations différentes d'un circuit programmable à l'autre.

Un exemple de réalisation d'un circuit de permutation utilisé dans l'invention est détaillé sur le schéma de principe de la figure 3.

Le bus de données DBUS ayant une largeur de n bits, le circuit de permutation comprend n étages de permutation, soit Ep0 à Ep7 dans l'exemple (n = 8).

Chaque étage de permutation reçoit en entrée les n bits du bus de données, soit D0-D7 dans l'exemple et délivre un bit de sortie débrouillé : l'étage de permutation Ep0 délivre le bit de sortie Di0, ... l'étage de permutation Ep7 délivre le bit de sortie Di7. Ce sont ces bits de sortie Di0-Di7 qui sont appliqués en entrée du registre d'instructions RI.

Chaque étage de permutation comprend n cellules de permutation c0-c7 dans l'exemple dont la fonction est celle d'un fusible, une seule parmi n permettant la transmission du bit de données d'entrée associé sur le bit de donnée de sortie de l'étage.

A chaque étage de permutation correspond un rang de cellule passante différent.

Dans l'exemple représenté, la cellule passante de l'étage

```
Ep0 est    c7
Ep1        c3

Ep2        c6
           .
           .
           .
Ep7        c0.
```

Le circuit de permutation DBr1 est ainsi caractérisé par une sélection de permutation pour chaque étage. Pour un bus de n bits de données, on a :

n possibilités pour le premier étage Ep0;

n-1 possibilités pour le deuxième étage Ep1;

et ainsi de suite.

Le procédé de brouillage selon l'invention permet ainsi n! possibilités différentes de brouillage pour un bus de données de largeur n bits.

Si les mémoires ont une largeur différente du bus de n bits données et notamment une plus grande largeur, chaque tranche de n bits subit la permutation selon l'invention. Et si la mémoire programme est plus large, le générateur de code exécutable devra effectuer le brouillage des données de programme exécutable après les avoir découpées par tranches de n bits.

Un exemple de réalisation d'un circuit de permutation selon l'invention est représenté en figures 4 et 5.

La figure 4 représente un dessin topologique de deux étages de permutation Ep0 et Ep1 à n = 8 bits d'entrée (D0-D7) et un bit de sortie, respectivement Di0 et Di1, et la figure 5 représente une coupe AA de ce dessin.

Pour chaque étage de permutation Ep0, Ep1 on a une couche d'interconnexion médiane DM0, DM1, connectée à la ligne de bit de sortie de l'étage.

Et, pour chaque cellule c0-c7 d'un étage de permutation, on a une couche d'interconnexion locale connectée à la ligne de bit d'entrée associée à cette cellule. Par exemple, on a une couche d'interconnexion locale DI0 pour la cellule c0, connectée à la ligne de bit de données d'entrée D0.

La programmation d'un étage de permutation consiste alors en une extension de la couche d'interconnexion médiane à la couche d'interconnexion locale d'une cellule parmi les n cellules de cet étage de permutation. Dans l'exemple représenté, on a ainsi pour l'étage Ep0 une extension de la couche d'interconnexion médiane DM0 à la couche d'interconnexion locale DI de la cellule c7 et pour l'étage Ep1, on a une extension de la couche d'interconnexion médiane DM1 à la couche d'interconnexion locale DI3 de la cellule c3.

Dans l'exemple représenté, les lignes de bit de données d'entrée D0-D7 sont en métal, et parallèles entre elles et la ligne de bit de données de sortie, par exemple Di0, est en polysilicium, dans un plan compris entre le plan des lignes de bit de données d'entrée et le substrat. Le contact de la couche d'interconnexion médiane à la ligne de bit de données de sortie se fait par une ligne de métal réalisée dans le même plan que les lignes de bit de données d'entrée et parallèle à celles-ci. Cette ligne de métal est notée Im0 pour l'étage Ep0 et Im1 pour l'étage Ep1.

Les couches d'interconnexion médianes ou locales peuvent être réalisées en diffusion, en métal ou autres, selon la technologie du circuit programmable.

Par exemple et comme plus particulièrement représenté en figures 4 et 5, elles sont réalisées en diffusion, par implantation ionique (appelé procédé d'implant). Ou bien la couche d'interconnexion médiane est elle en métal.

En coupe AA représentée sur la figure 5, on distingue par exemple pour l'étage Ep0 la ligne de sortie de données en polysilicium Di0, connectée à la ligne de métal Im0 au-dessus, et les contacts de cette ligne de métal sur la couche d'interconnexion médiane DM0 de l'étage de permutation Ep0.

Les couches d'interconnexions locales DI3 et DI4 des cellules c3 et c4 des étages Ep0 et Ep1 sont aussi visibles sur cette figure 5. Et, on voit l'extension de la couche d'interconnexion médiane DM1 à la couche d'interconnexion locale DI3 de la cellule c3 de l'étage Ep1.

La particularisation d'un étage de permutation se fait donc par extension de la couche d'interconnexion médiane à la couche d'interconnexion locale qui contacte la ligne de bit de donnée d'entrée sélectionnée (dessin du masque).

La programmation du circuit de permutation selon l'invention, consiste alors selon ce procédé de fabrication à sélectionner l'extension de couche d'interconnexion médiane à la couche d'interconnexion locale qui définit la permutation, extension qui va permettre de relier la ligne de bit d'entrée à la ligne de bit de sortie.

Cette particularisation programmable est particulièrement aisée à mettre en oeuvre, permettant une souplesse d'adaptation du procédé de fabrication à chaque circuit programmable : utilisation de circuits de permutation sur les autres mémoires ou non, avec identité de brouillage ou non, etc... et permettant un brouillage différent pour chaque circuit programmable : On n'a plus d'identité de protection au sein d'une même famille. Il en résulte une protection considérablement améliorée des circuits programmables.

Avantageusement, pour une mémoire programme de type mémoire morte, on utilise le même procédé de fabrication que pour le circuit de permutation :

- d'une part, on réalise les cellules de permutation et les cellules de mémoire morte dans les mêmes étapes de fabrication, avec les mêmes masques, au moyen des mêmes types de couches d'interconnexion; la programmation ou particularisation du circuit pour le programme et le brouillage spécifiés par le client se fait au même niveau de fabrication, d'où une grande simplification de la fabrication;

- d'autre part, on augmente la difficulté d'inspection visuelle du circuit par utilisation de technologies similaires.

Selon le procédé décrit en relation avec les figures 4 et 5, la programmation des données exécutables brouillées en mémoire morte est réalisée, en utilisant une couche d'interconnexion médiane pour une même ligne de bit et une couche d'interconnexion locale pour chaque cellule étendue ou non à la couche d'interconnexion médiane selon l'état programmé voulu 0 ou 1 de la cellule.

Revendications

1. Procédé de brouillage numérique de données, caractérisé en ce que le brouillage est réalisé par un générateur de code exécutable qui effectue une discrimination des données d'instructions parmi toutes les données d'un programme, pour appliquer une première permutation de bits ou bien seulement à ces données d'instructions, ou bien seulement aux autres données.

2. Procédé de brouillage numérique selon la revendication 1, le code exécutable obtenu étant chargé dans un circuit programmable, en mémoire programme reliée par au moins un bus de données à une unité de commande, caractérisé en ce que les données d'instructions transitent brouillées sur le bus de données, un débrouillage de la première permutation leur étant appliqué en entrée d'instructions de l'unité de commande.

3. Procédé de brouillage numérique selon la revendication 1 ou 2, pour un circuit programmable comprenant au moins une mémoire réinscriptible de la mémoire programme ou non, caractérisé en ce qu'il consiste à appliquer les données de cette mémoire réinscriptible, à un circuit de permutation placé entre ladite mémoire et le bus de données, pour effectuer un brouillage des données à mémoriser et un débrouillage des données à transmettre, correspondant à une deuxième permutation.

4. Procédé de brouillage numérique selon la revendication 3, pour un circuit programmable comprenant au moins deux mémoires réinscriptibles, caractérisé en ce que leur circuit de permutation associé effectue des permutations différentes.

5. Procédé de brouillage numérique selon la revendication 2, pour une première permutation dans le générateur appliquée aux autres données que les données d'instructions, caractérisé en ce qu'il consiste à appliquer aux données de mémoire programme une permutation inverse de la première entre la mémoire programme et le bus de données, en sorte que les données d'instructions transitent brouillées sur le bus de données.

6. Circuit programmable comprenant au moins un circuit de permutation pour mettre en oeuvre le procédé de brouillage selon l'une des revendications 2 à 5, caractérisé en ce que le circuit de permutation comprend n étages de permutation à n bits de données d'entrée (D0-D7) et un bit de donnée de sortie (Di0), chaque étage comprenant n cellules de permutation, une par bit de données d'entrée.

7. Procédé de fabrication d'un circuit de permutation comprenant n étages de permutation à n bits de données d'entrée (D0-D7) et un bit de donnée de sortie (Di0), chaque étage comprenant n cellules de permutation, une par bit de données d'entrée, caractérisé en ce que

   - pour chaque étage de permutation (Ep0), une couche d'interconnexion médiane (DM0) est connectée à une ligne de bit de sortie (Di0) et

   - pour chaque cellule (c0) de cet étage de permutation, une couche d'interconnexion locale (Dl0) est connectée à la ligne de bit d'entrée correspondante (D0),

   la programmation d'un étage de permutation (Ep0) consistant en une extension de la couche d'interconnexion médiane (DM0) dudit étage de permutation à la couche d'interconnexion locale d'une cellule parmi les n cellules dudit étage de permutation.

8. Procédé de fabrication selon la revendication 7, caractérisé en ce que la couche d'interconnexion médiane et les couche d'interconnexions locales sont des couche d'interconnexions réalisées par implantation ionique.

9. Procédé de fabrication selon la revendication 7, caractérisé en ce que la couche d'interconnexion médiane est réalisée en métal et les couches d'interconnexions locales en diffusion, par implantation ionique.

10. Procédé selon la revendication 7, 8 ou 9, caractérisé en ce que les mêmes types de couches d'interconnexion permettent la programmation d'une mémoire morte et des circuits de permutation.

FIG_1

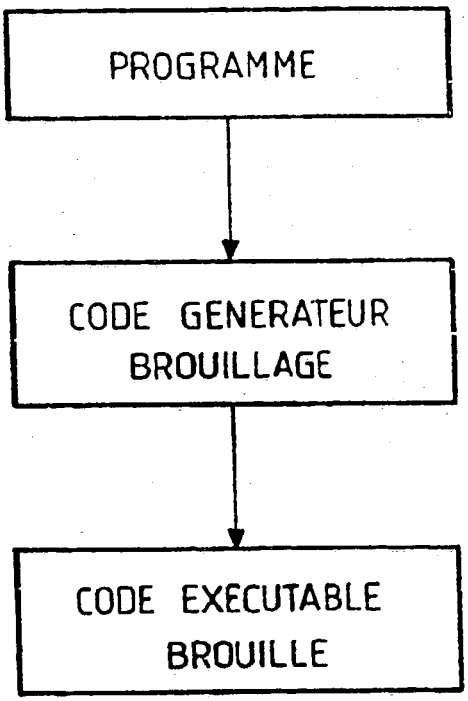

PROGRAMME
CODE GENERATEUR BROUILLAGE
CODE EXECUTABLE BROUILLE

FIG_2

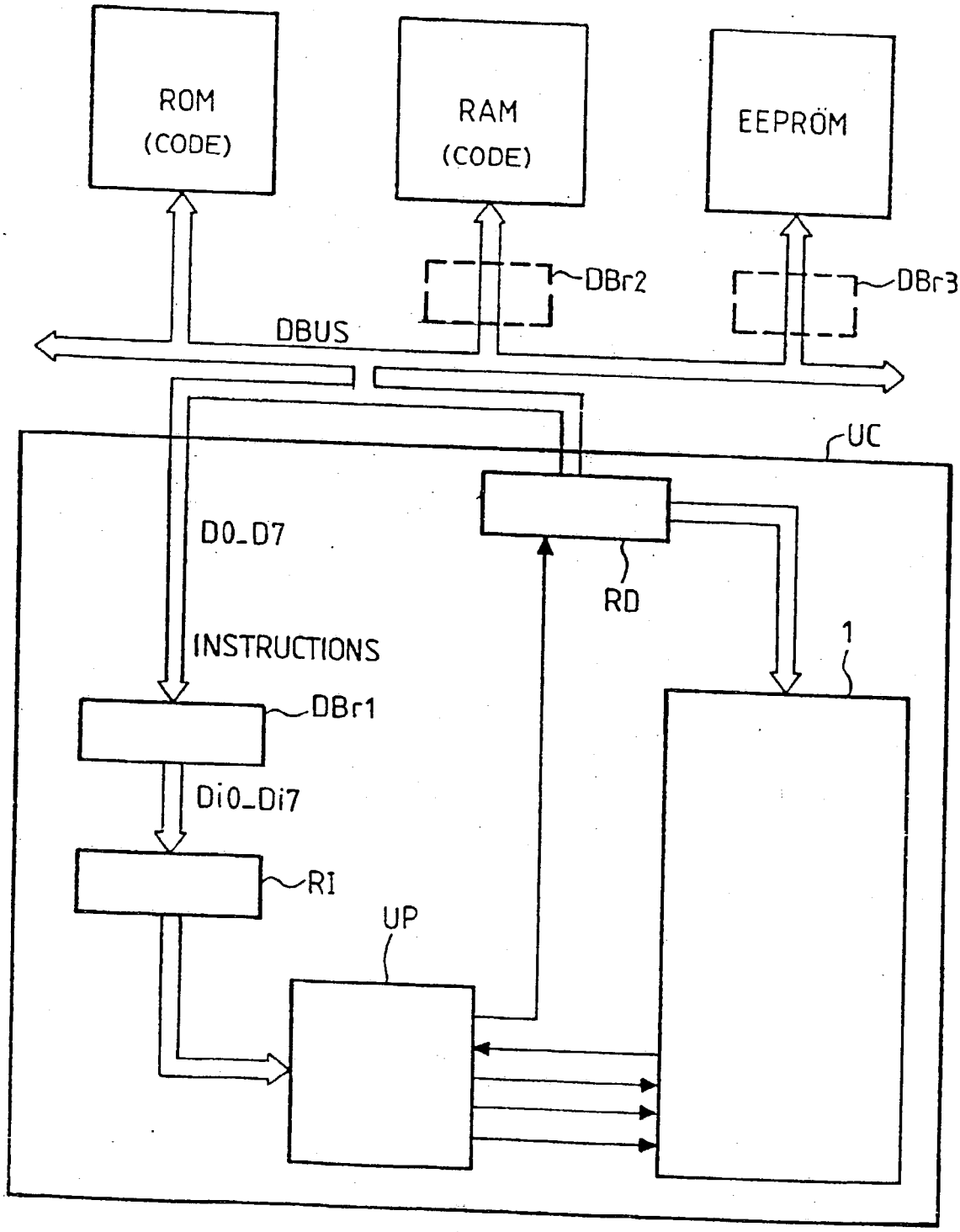
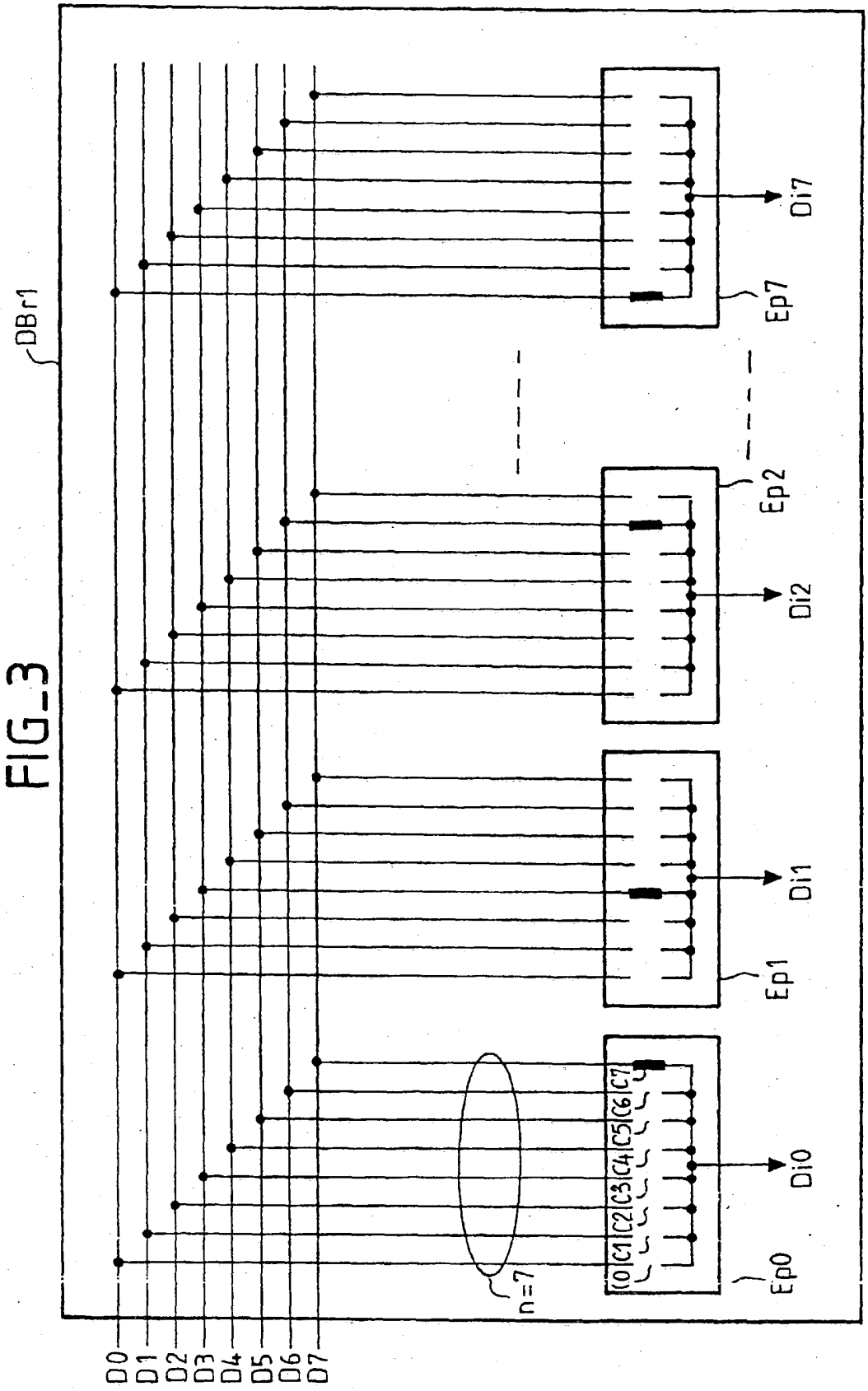

ROM
(CODE)
RAM
(CODE)
EEPROM
DBr2
DBr3
DBUS
UC
D0_D7
RD
INSTRUCTIONS
1
DBr1
Di0_Di7
RI
UP

FIG_3


DBr1
D0
D1
D2
D3
D4
D5
D6
D7
n=7
C0
C1
C2
C3
C4
C5
C6
C7
Ep0
Ep1
Ep2
Ep7
Di0
Di1
Di2
Di7

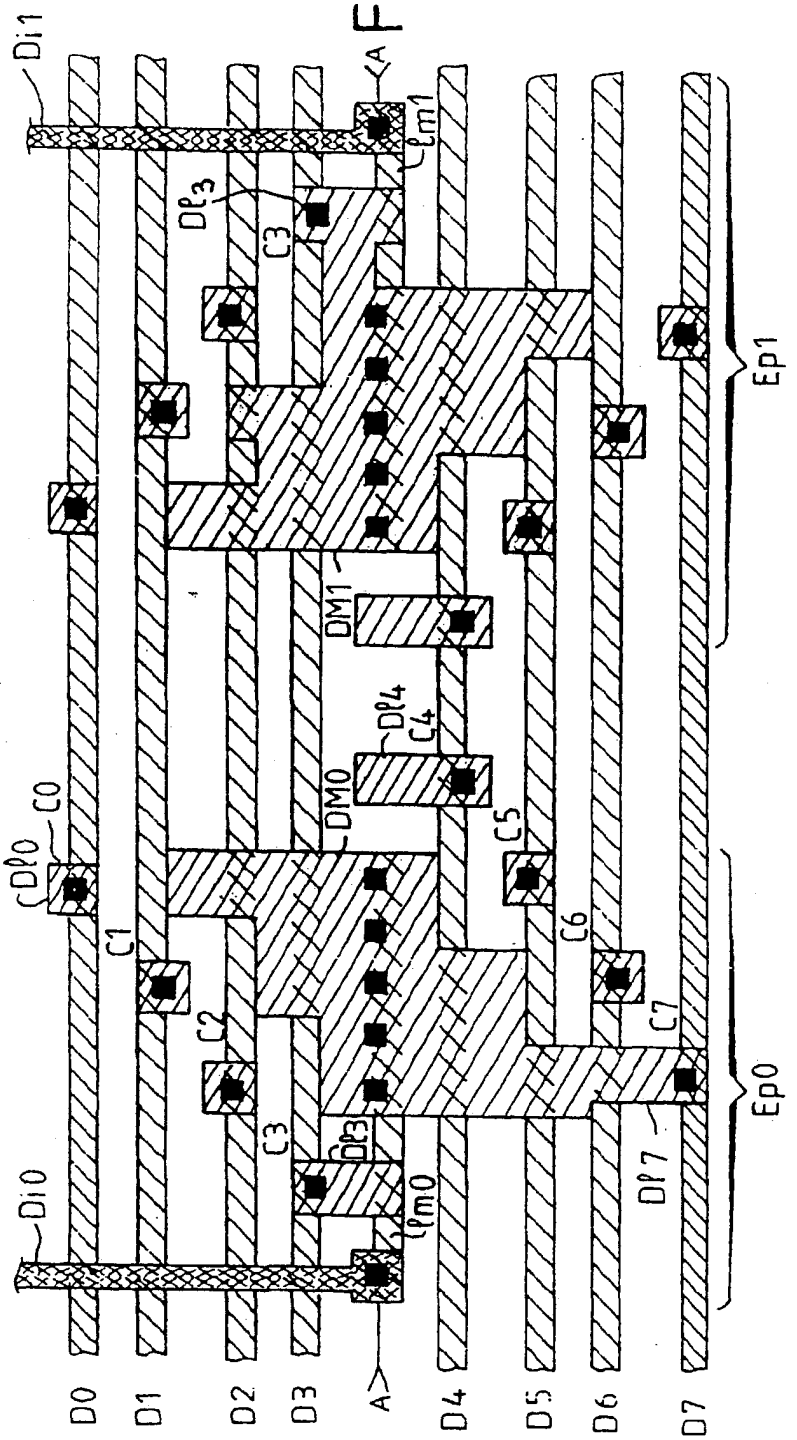

Di1
Di0
C0
C1
C2
C3
C3
C4
C5
C6
C7
DM1
DM0
Ep1
Ep0
A
A
D0
D1
D2
D3
D4
D5
D6
D7

FIG_4

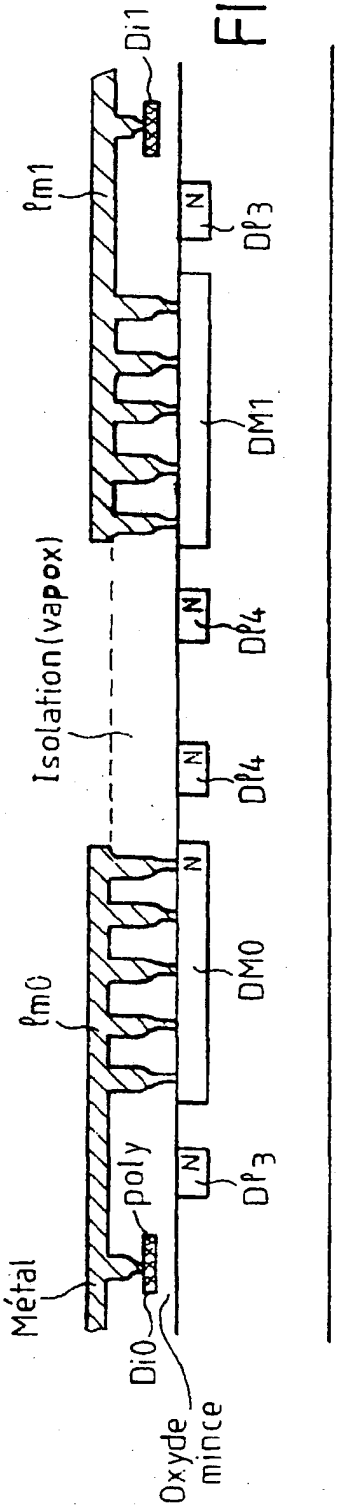

Di1
Isolation(vapox)
DM1
DM0
poly
Métal
Oxyde mince
Di0

FIG_5

Office européen des brevets

RAPPORT DE RECHERCHE EUROPEENNE

Numero de la demande
EP 95 40 1789

DOCUMENTS CONSIDERES COMME PERTINENTS

Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes	Revendication concernée	CLASSEMENT DE LA DEMANDE (Int.Cl.6)
X Y	US-A-4 633 388 (CHIU) * le document en entier *	1 2,3,6	G06F12/14 G06F7/00 H01L23/535
Y	US-A-4 278 837 (BEST) * abrégé; figures 1-3,10 * * colonne 2, ligne 34 - colonne 3, ligne 2 * * colonne 5, ligne 11 - ligne 47 * * colonne 13, ligne 1 - ligne 63 *	2	
Y	US-A-5 095 525 (ALMGREN ET AL) * abrégé; figures 6,7B * * colonne 9, ligne 20 - ligne 26 *	3,6-10	
Y	FR-A-2 504 730 (EFCIS) * page 1, ligne 1 - page 4, ligne 14 *	7-10	
Y	IBM TECHNICAL DISCLOSURE BULLETIN, vol.33, no.1B, Juin 1990, NEW YORK, US; pages 323 - 324 'Cell Design for Multiple Logic Circuit Families' * le document en entier *	10	

DOMAINES TECHNIQUES RECHERCHES (Int.Cl.6)
G06F G11C H01L

Le présent rapport a été établi pour toutes les revendications

Lieu de la recherche	Date d'achèvement de la recherche	Examinateur
LA HAYE	10 Novembre 1995	Powell, D

CATEGORIE DES DOCUMENTS CITES

X : particulièrement pertinent à lui seul
Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie
A : arrière-plan technologique
O : divulgation non-écrite
P : document intercalaire

T : théorie ou principe à la base de l'invention
E : document de brevet antérieur, mais publié à la date de dépôt ou après cette date
D : cité dans la demande
L : cité pour d'autres raisons

& : membre de la même famille, document correspondant

EPO FORM 1503 03.82 (P04C01)

Method for numerically scrambling data and its application to a programmable circuit

Patent Number: US5850452
Publication date: 1998-12-15
Inventor(s): SOURGEN LAURENT (FR); WUIDART SYLVIE (FR)
Applicant(s): ST MICROELECTRONICS SA (IT)
Requested Patent: EP0694846, B1
Application Number: US19950509363 19950731
Priority Number(s): FR19940009485 19940729
IPC Classification: H04L9/00
EC Classification: G06F1/00N1C, G06F7/00C
Equivalents: DE69526753D, FR2723223, JP8123680

Abstract

The present invention concerns a method for the numerical scrambling by permutation of data bits in a programmable circuit comprising a control unit and at least one data bus (DBUS) to transmit data between the control unit and several memory circuits. It consists of having data on the bus either in a scrambled form or in an unscrambled form according to whether it is instructions data or not. And data in some of the memories is scrambled. The present invention also concerns a method for realising a permutation circuit.

DOCKET NO: P2001,0019
SERIAL NO: _____
APPLICANT: Berndt Gammel et al.

LERNER AND GREENBERG P.A.
P.O. BOX 2480
HOLLYWOOD, FLORIDA 33022
TEL. (954) 925-1100